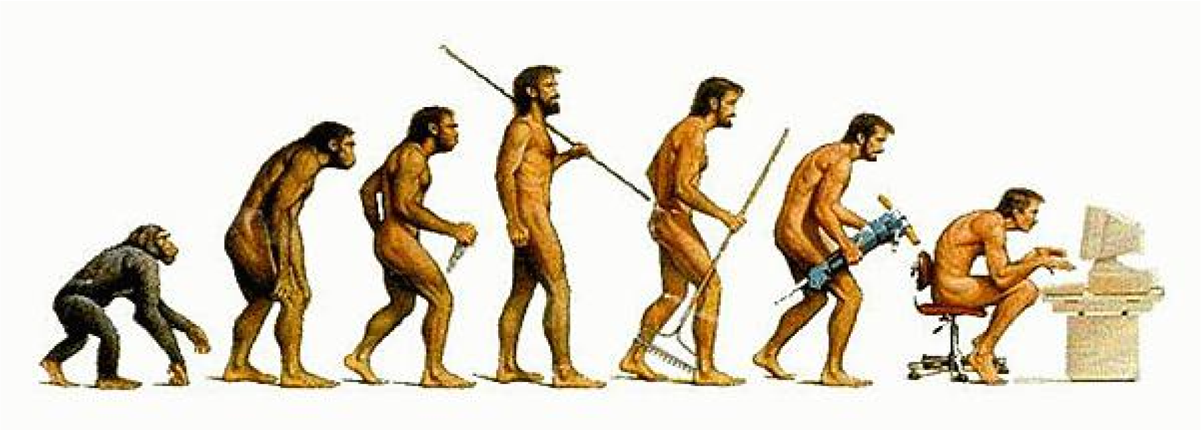
COMPUTATIONAL PHYSICS

Shuai Dong

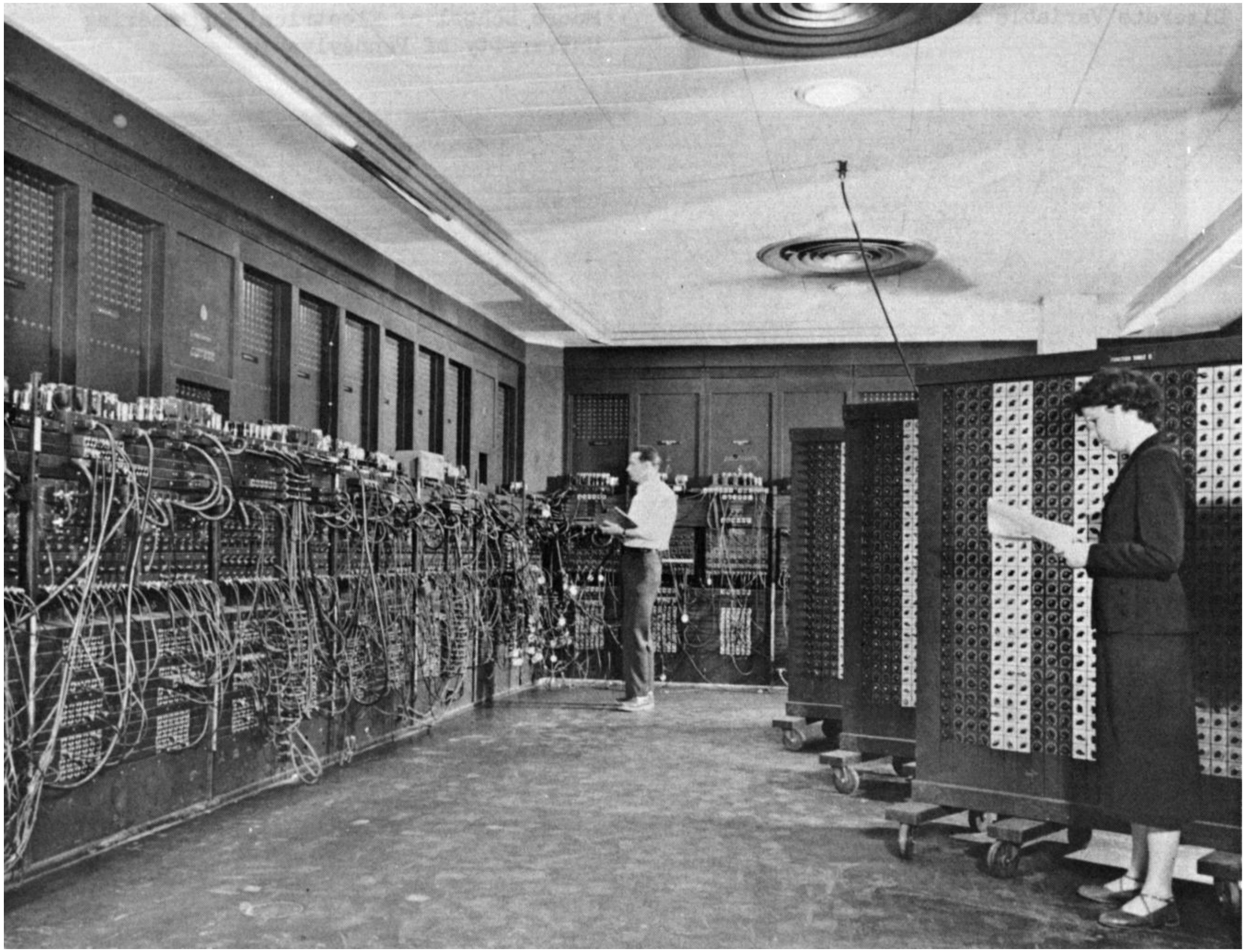


Evolution: Is this our final end-result?

Outline

- Brief history of computers
- Supercomputers
- Brief introduction of computational science
- Some basic concepts, tools, examples

Birth of Computational Science (Physics)



The first electronic general-purpose computer:

Constructed in Moore School of Electrical Engineering, University of Pennsylvania, 1946

ENIAC: Electronic Numerical Integrator And Computer

ENIAC

Electronic Numerical Integrator And Computer

- Design and construction was financed by the United States Army.
- Designed to calculate artillery firing tables for the United States Army's Ballistic Research Laboratory.

- It was heralded in the press as a "Giant Brain".
- It had a speed of one thousand times that of electro-mechanical machines.

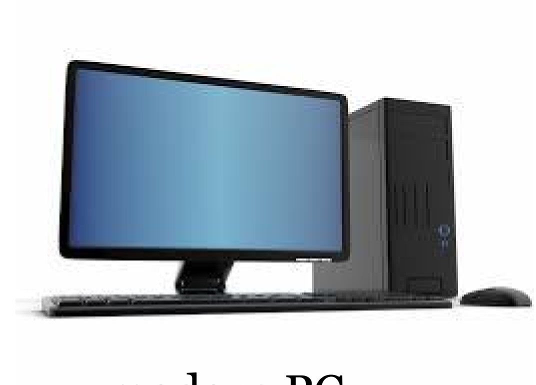
- ENIAC was named an IEEE Milestone in 1987.

Gaint Brain

- ENIAC contained 17,468 vacuum tubes, 7,200 crystal diodes, 1,500 relays, 70,000 resistors, 10,000 capacitors and around 5 million hand-soldered joints. It weighed more than 27 tons, took up 167 m^2, and consumed 150 kW of power.
- This led to the rumor that whenever the computer was switched on, lights in Philadelphia dimmed.
- Input was from an IBM card reader, and an IBM card punch was used for output.

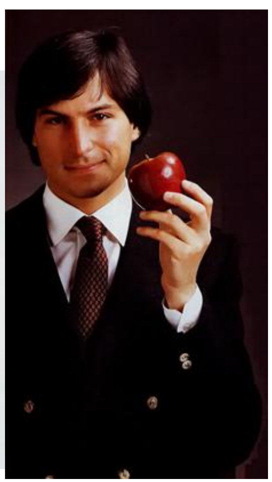
Development of *micro*-computers

1981 IBM PC 5150
CPU: 8088, 5 MHz
Floppy disk or cassette



modern PC
CPU: Intel i3,i5,i7,
3 GHz
Solid state disk

1984 Macintosh

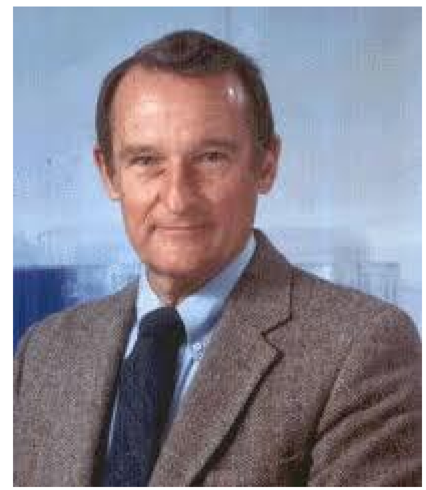


Steve Jobs

modern iMac

Supercomputers

The CDC (Control Data Corporation) 6600, released in 1964, is generally considered the first supercomputer.



Seymour Roger Cray (1925-1996)
The father of supercomputing, who created the supercomputer industry.

Cray Inc.

Cray I 1976-1981
Cray II 1985-1989
……
Cray XT5 2009-2010
Cray XK7 2012-2013

Cray-2, the fastest computer between 1985 and 1989.

Top 10 supercomputers

- IBM Power System: Summit(1), Sierra (2), Lassen (10)
- NRCPC: Sunway TaihuLight (3)
- NUDT: Tianhe 2A (4)
- Dell: Frontera (5)
- Cray series: Piz Daint (6), Trinity (7)
- Fujitsu： ABCI (8)
- Lenovo: SuperMUC-NG (9)

http://www.top500.org/

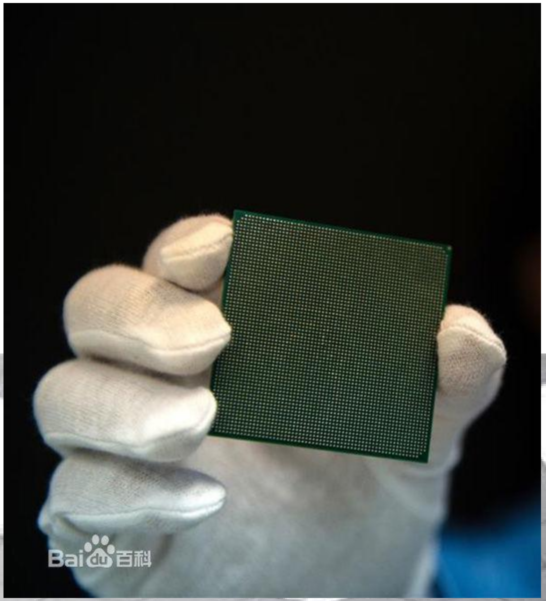
Rank	Site	System	Cores	Rmax	Rpeak	Power (kW)
1	Oak Ridge National Laboratory	IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband	2414592	148600	200795	10096
2	Lawrence Livermore National Laboratory	IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband	1572480	94640.0	125712	7438
3	National Supercomputing Center in Wuxi	Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45 GHz, Sunway	10649600	93014.6	125436	15371
4	National Super Computer Center in Guangzhou	Tianhe-2 - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.2 GHz, TH Express-2, Intel Xeon Phi 31S1P	3120000	33862.7	54902.4	17808
5	Texas Advanced Computing Center/Univ. of Texas	Frontera - Dell C6420, Xeon Platinum 8280 28C 2.7GHz, Mellanox InfiniBand HDR	448448	23516.4	38745.9	

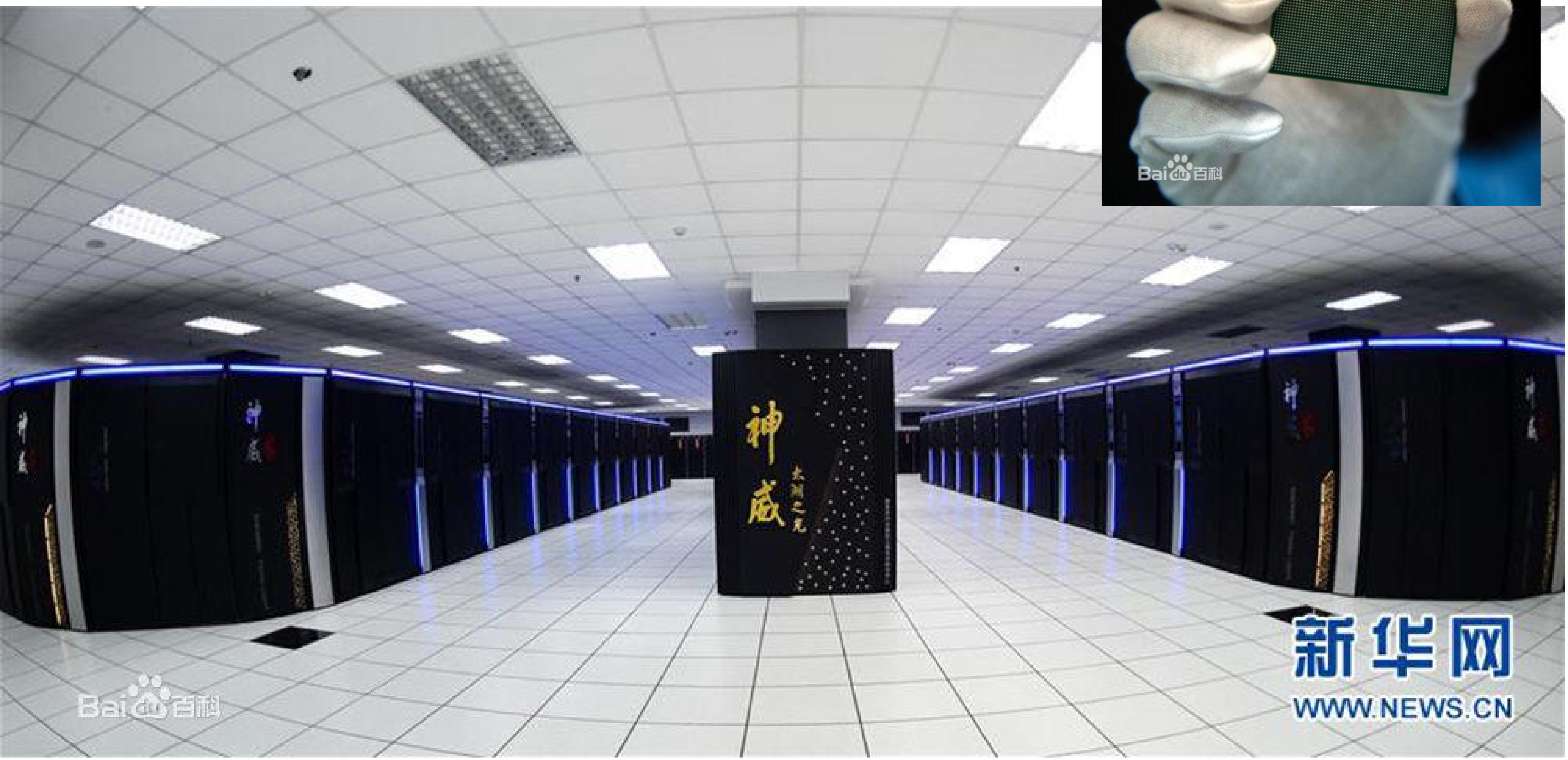
Supercomputer- Summit, since 2018

- Sponsor DOE Manufacturer IBM
- Location Oak Ridge National Laboratory
- Nodes: 4,608 Processor: IBM POWER9™ (2/node)
- GPUs: 27,648 NVIDIA Volta V100s (6/node)
- Memory/node: 512GB DDR4 + 96GB HBM2
- NV Memory/node: 1600GB
- Total System Memory: >10PB DDR4 + HBM + Non-volatile
- Interconnect Topology: Mellanox EDR 100G InfiniBand, Non-blocking Fat Tree

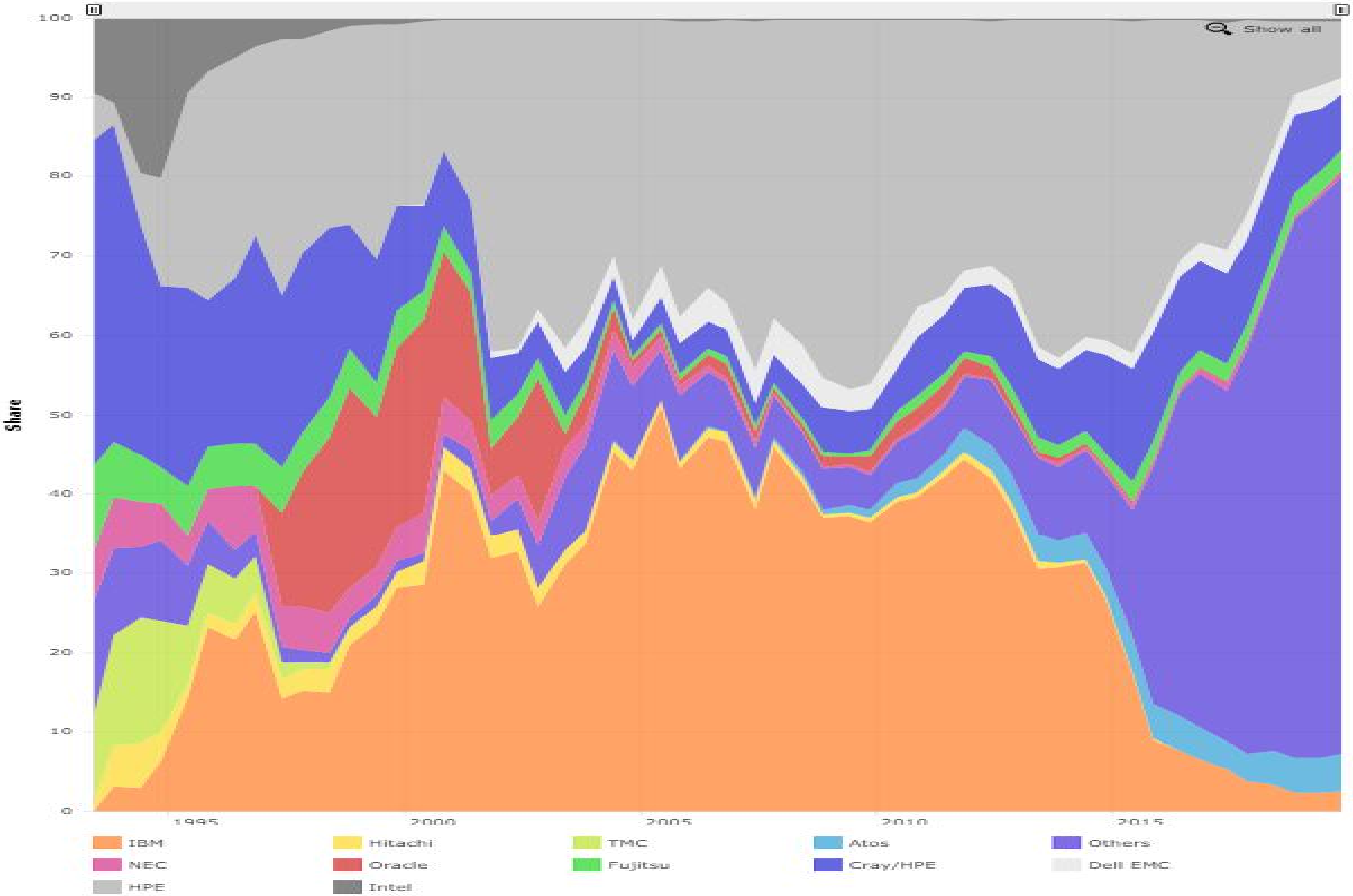
Sunway-Taihulight

- 40,960 CPUs, 10,649,600 cores
- originated from DEC Alpha 21164
- Reduced Instruction Set Computer (RISC)
- 1.5GHz



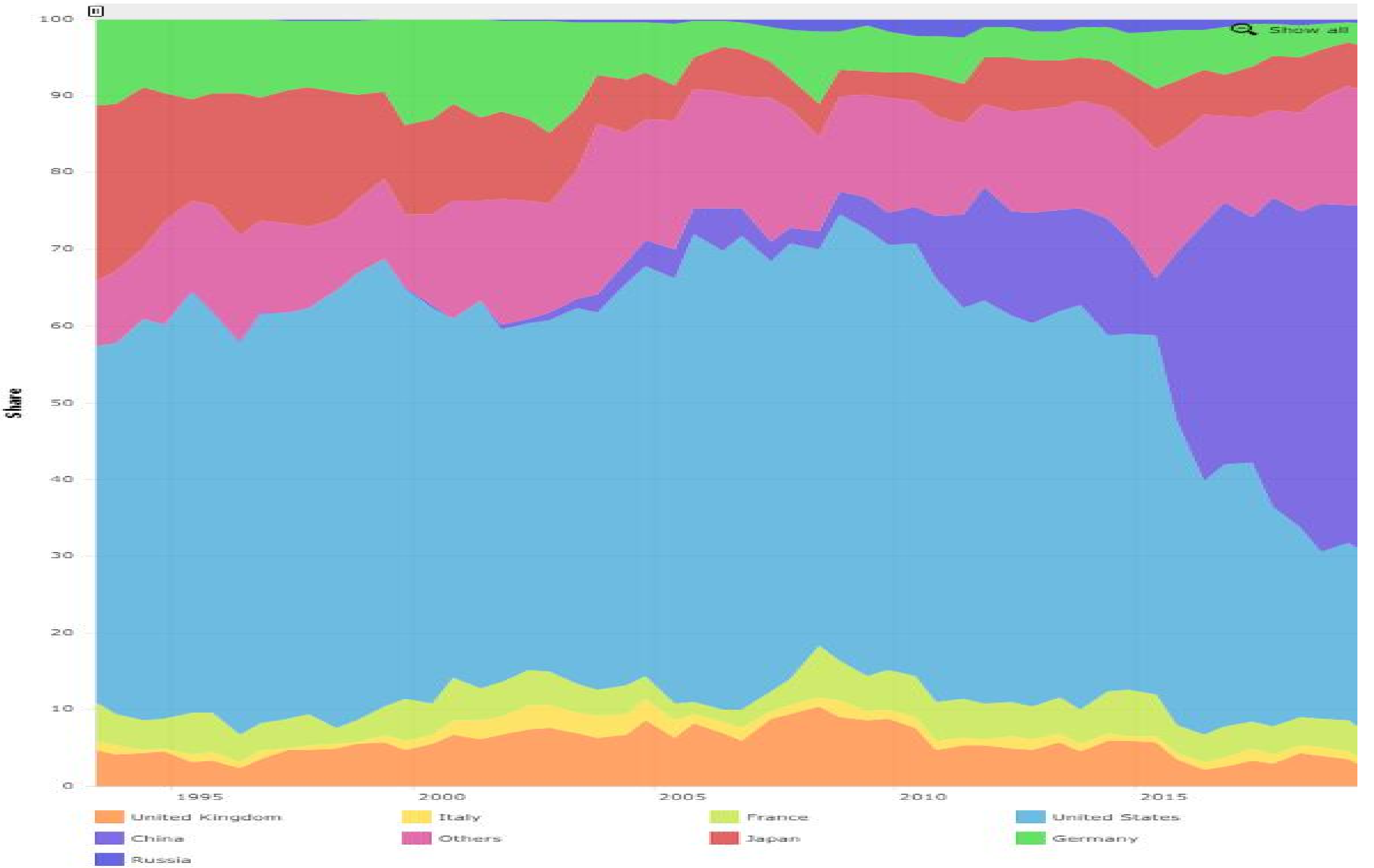


Tendency



Show all
Share
100
90
80
70
60
50
40
30
20
10
0
1995
2000
2005
2010
2015
IBM
Hitachi
TMC
Atos
Others
NEC
Oracle
Fujitsu
Cray/HPE
Dell EMC
HPE
Intel

Tendency



Show all
Share
100
90
80
70
60
50
40
30
20
10
0
1995
2000
2005
2010
2015
United Kingdom
Italy
France
United States
China
Others
Japan
Germany
Russia

Applications of Computational Science

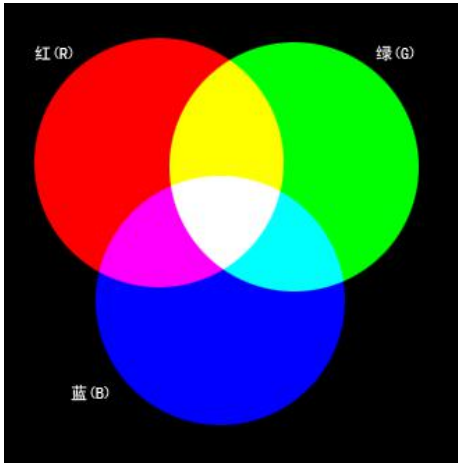
- Dynamics of the global environment, weather forecasting
- Molecular dynamics simulation; design of drugs, materials and devices
- Simulation of weapons
- Cartoon, 3D movies, multimedia
- Simulation of finance, traffic, society ......
- ......

The base of all above applications is the computational physics!

What will be studied?

- A cup of cocktail:

- Computer science
- Mathematics
- Physics



It is not the most important course,
but maybe the most useful course!

The aim of this course

To Learn how to establish efficient and accurate algorithms as well as how to write clean and structured computer programs for most problems encountered in physics and related fields.

Syllabus of this course

1. Introduction of computational physics
2. Approximation of functions
3. Numerical calculus
4. Ordinary differential equations
5. Numerical methods for matrices
6. Spectral analysis
7. Molecular dynamics simulations
8. Monte Carlo simulations
9. High-performance computing
10. A comprehensive project

http://physics.seu.edu.cn/sdong

Tools: computer languages

- High-level programming languages, e.g. C/C++/Fortran/Java.....
- I use C, with a few C++ options (mostly for input/output).
- The object-oriented (OO) feature of C++ is not very useful in the computational physics. Maybe it is useful to desigin system softwares, but not popular in science.

A code example

```
#include <iostream.h>

void main()
{
    cout<<"Hello, Computational Physics!"<<endl;
}
```

1.1.hello.cpp

A code example

```
• #include <iostream>

• using namespace std;

• int main()
• {
      cout<<"Hello, Computational Physics!"<<endl;
      return 0;
• }
```

1.2.hello.cpp

Compiler

- A translator, a program called a compiler, is used to translate (or compile) the code to produce an executable file in binaries.

Human readable
e.g. .c/.cpp
cout<<"hello, world!"<<endl;

Computer "readable"
e.g. .exe
1010110110100001110101

- Most compilers also have an option to produce an objective file first and then link it with other objective files and library routines to produce a combined executable file.

Human readable
e.g. .c/.cpp

Computer "readable"
e.g. .obj/.dll/.so/.a
dll: Dynamic Link Library

Some modern C/C++ compilers

- gcc/g++: GNU Compiler Collection (9.2)
- icc: Intel® Parallel Studio XE 2020
- VC: Microsoft Visual Studio 2019

IDE: Integrated development environment

Orwell Dev-C++ (5.11), an IDE with g++ (4.9.2)

http://en.wikipedia.org/wiki/List_of_compilers

GCC/G++

- To bootstrap the GNU Operating System, Stallman asked Tanenbaum, the author of the Amsterdam Compiler Kit (also known as the Free University Compiler Kit) if he could use that software for GNU. When Tanenbaum told him that while the Free University Compiler Kit was free, the compiler was not, Stallman decided to write his own.

Richard Stallman
President of the Free Software Foundation
Stallman launched the GNU project in 1983 to create a Unix-like computer operating system composed entirely of free softwares.

National Academy of Engineering 2002

https://stallman.org/

GNU: GNU Not Unix

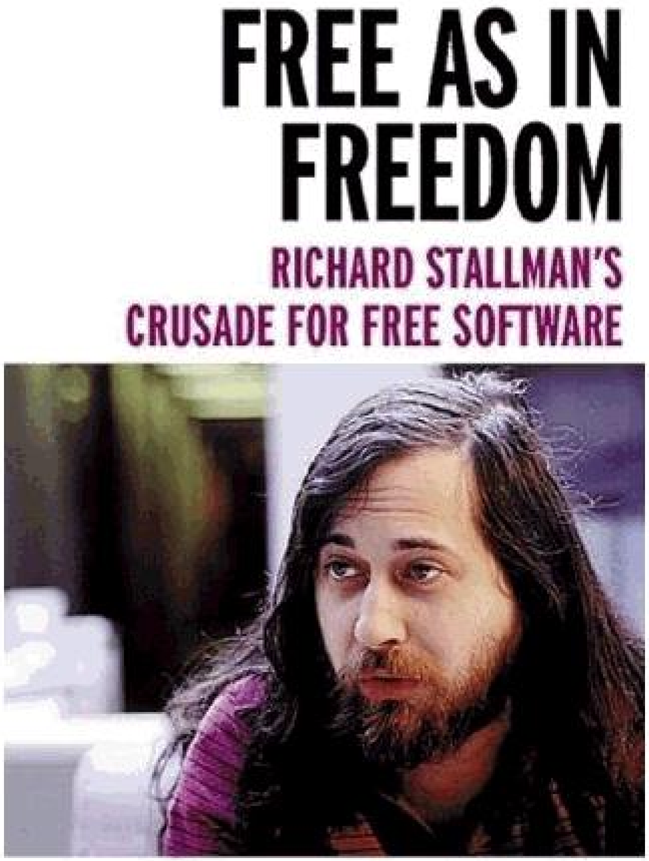
Free software:
Free as in freedom, not in free beer.

Richard Stallman

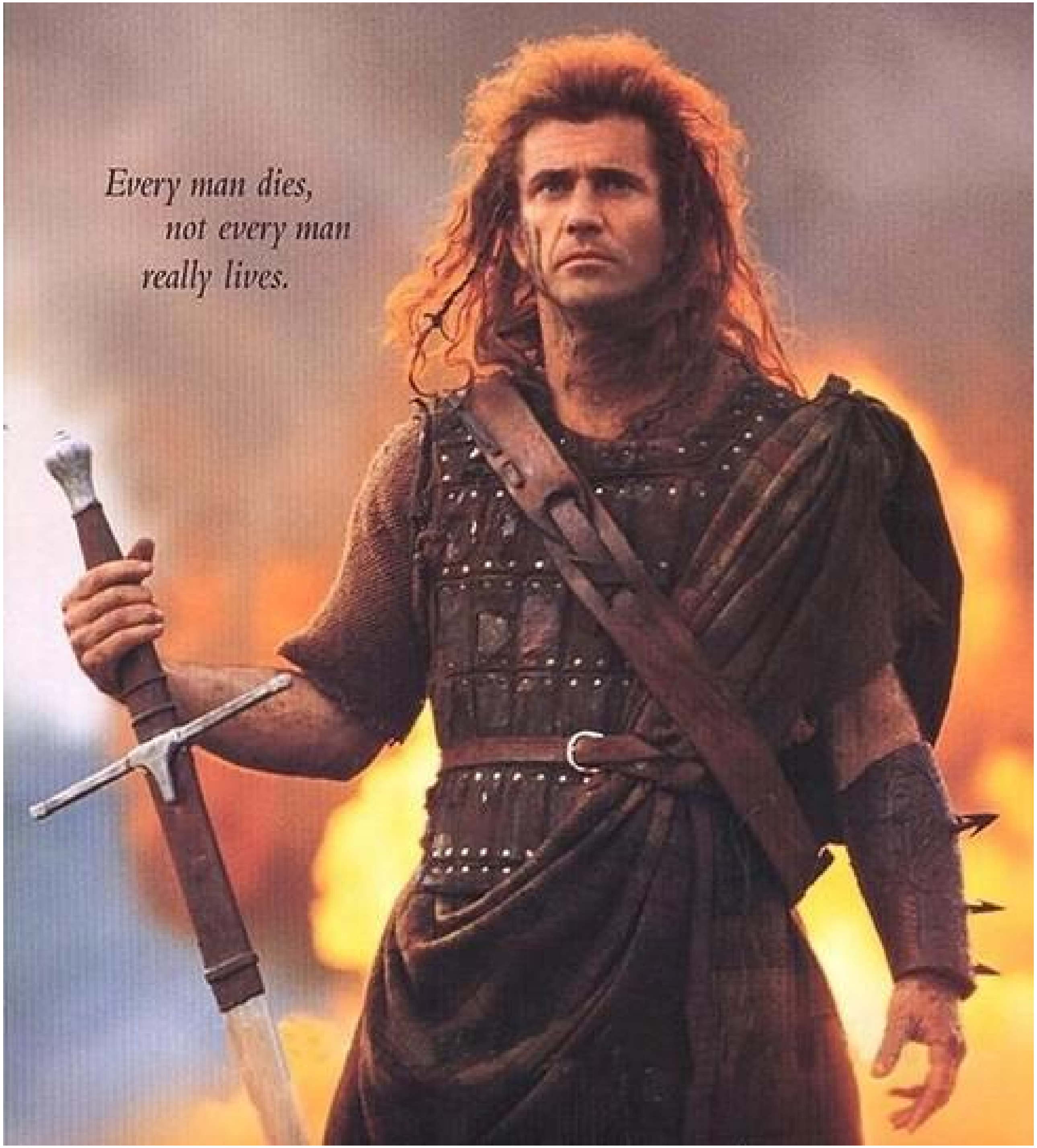
Copyright vs Copyleft

GPL: General Public License

www.gnu.org



Laptop made by Lemote powered by Loongson CPU



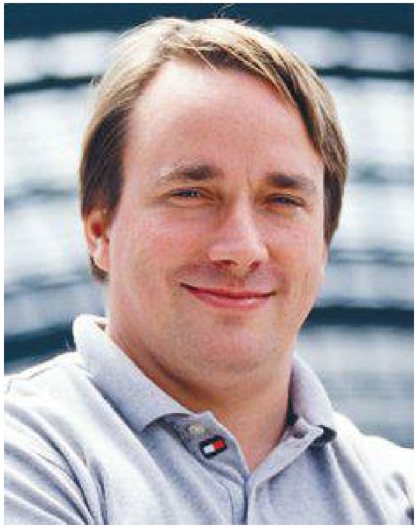
Brave Heart

Freedom!!!

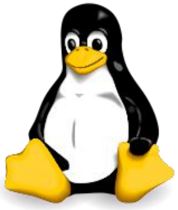
Operating systems

- GNU/Hurd (1990-now)
- now: 0.9 (2016)

- GNU/Linux (1991-now)
- 1991: 0.01
- 1994: 1.0
- 1996: 2.0
- 2011: 3.0
- 2015: 4.0
- 2019: 5.0
- now: 5.5



Linus Torvalds



Tux,
mascot of Linux

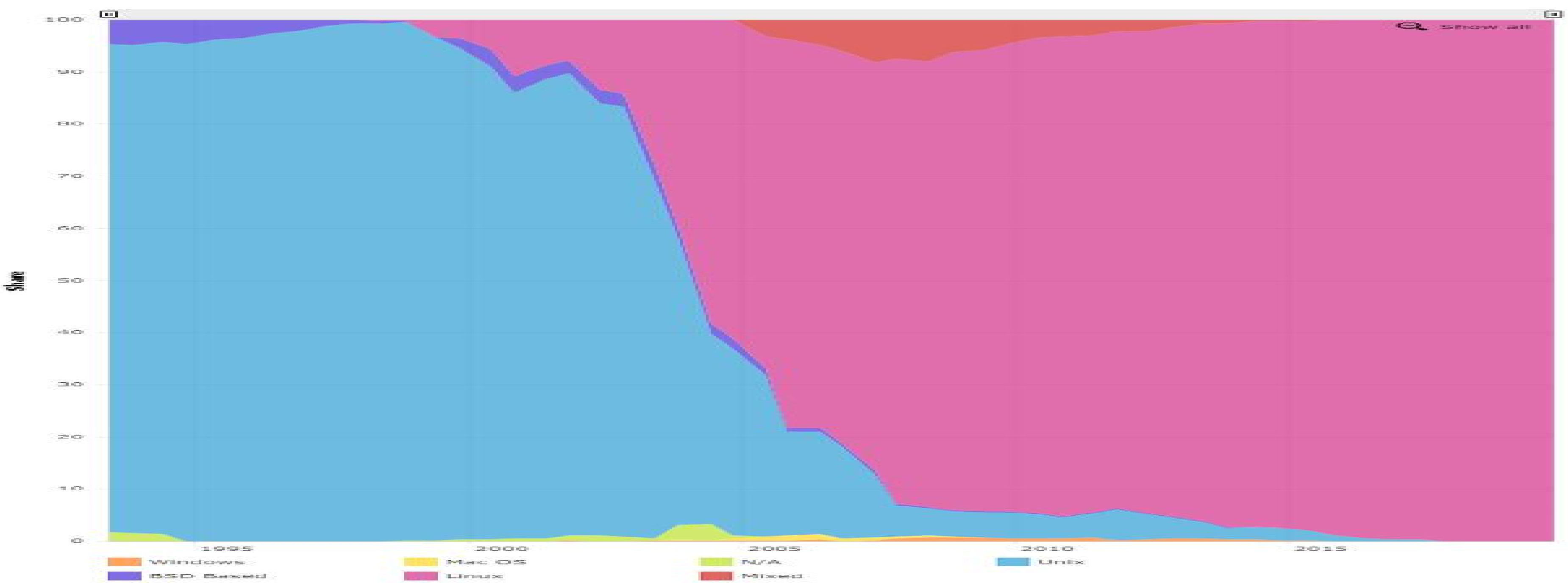
https://www.kernel.org/

Operating systems

- Linux - the dominant OS for supercomputers.
- Also works excellent for PCs/laptops.

Source	Linux	Unix	Mixed	Windows	BSD based	Mac OS
TOP500	100.0%	0.0%	0%	0%	0%	0%

Nov. 2019



Linux/Unix distributions

Many choices

Andriod's kernel is also Linux.

My favorite:
Debian
The Universal OS

Debian/Linux
Debian/FreeBSD
Debian/Hurd

http://www.debian.org

Compile a code

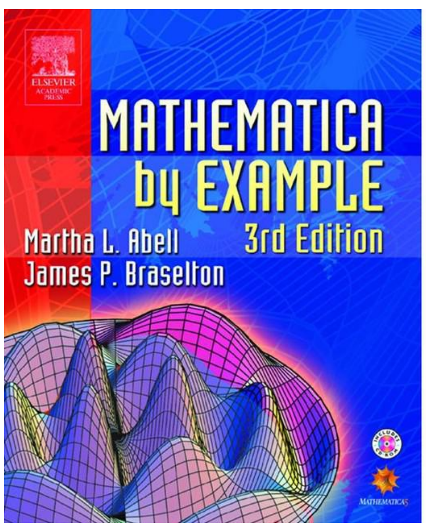
- In Linux, we can use the gcc/g++ command.
- g++ yourcode
- g++ -O3 yourcode -o a.out -Wall
- ./a.out

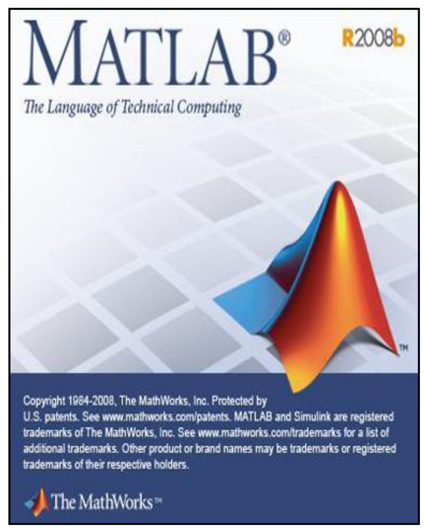
- In windows, we can use Dev-C++, which is an Integrated development environment (IDE) with the g++ as the compiler.

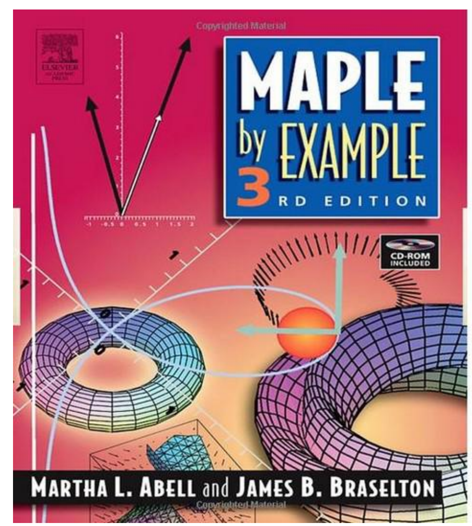
1.1 hello.cpp vs 1.2.hello.cpp

Other softwares

- Mathematica/Matlab/Maple ….
- Easy to use but expensive
- Also some free softwares on the Linux platform







Computer algorithms

A common process: to model the physical problem into a mathematic problem, the use computer to solve it.

A simple example: movement of a particle:

Newton's 2nd law equation (physics):

$$f = ma = m\frac{dv}{dt}$$

m: mass; a: acceleration, v: velocity, t: time

- small, equal intervals $\tau = t_{i+1} - t_i$

$$v_i \approx \frac{x_{i+1} - x_i}{t_{i+1} - t_i} = \frac{x_{i+1} - x_i}{\tau}$$

$$a_i \approx \frac{v_{i+1} - v_i}{t_{i+1} - t_i} = \frac{v_{i+1} - v_i}{\tau}$$

$$x_{i+1} = x_i + \tau v_i$$

$$v_{i+1} = v_i + \frac{\tau}{m} f_i$$

differential equations in continuous space
----> difference equations in discrete space

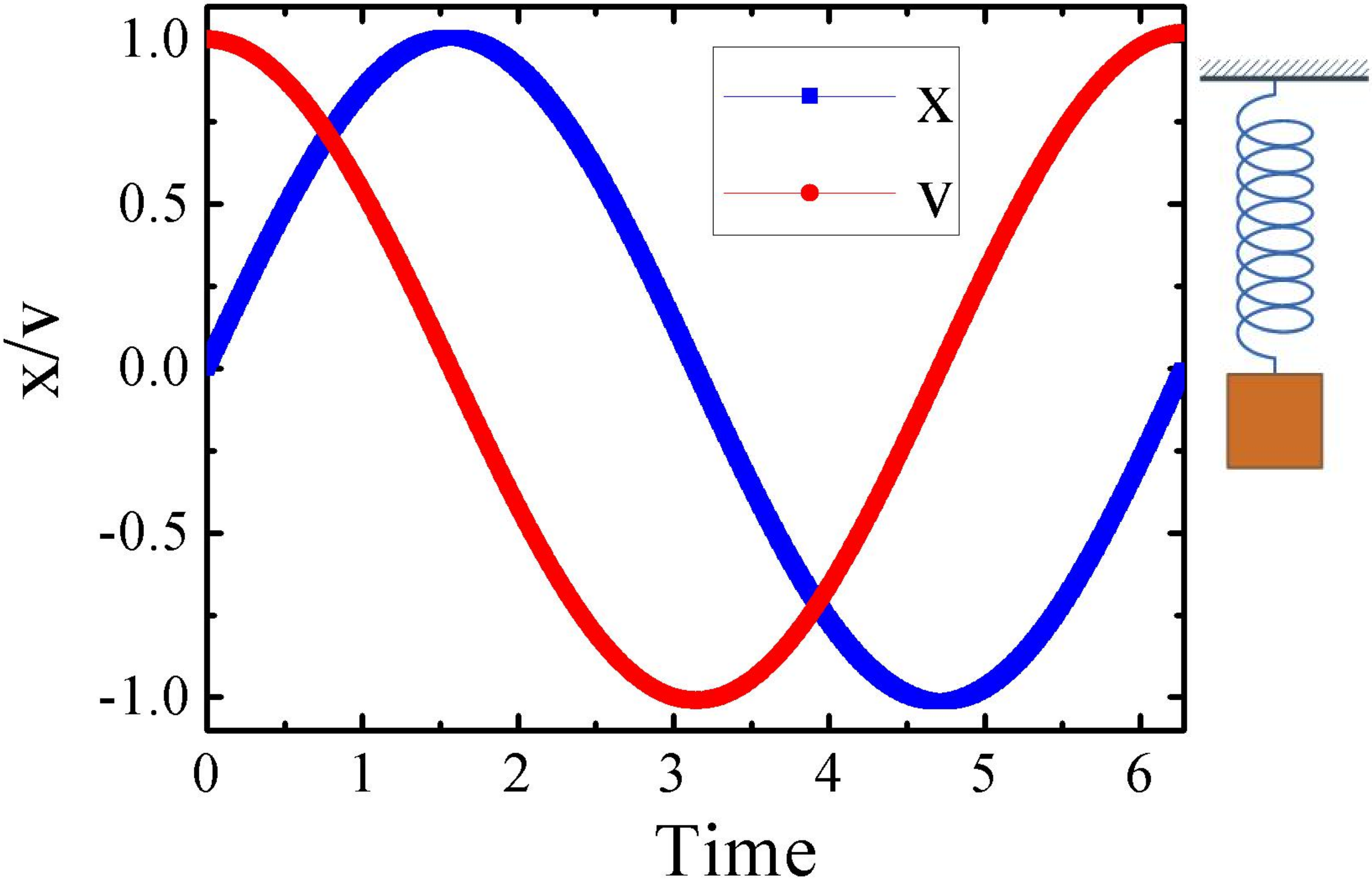
- If the initial position x_o and velocity v_o of the particle are given and the corresponding quantities at some later time are sought (the initial-value problem).

- We can obtain them recursively from the algorithm given in aforementioned equations. This algorithm is commonly known as the Euler method for the initial-value problem.

```
#include <iostream>
#include <cmath>

using namespace std;

int main()
{
    const int N=1000;                    //divide a period into N steps
    const double dt=2*M_PI/N;            //one step
    double x[N], v[N];                   //position, velocity

    x[0] = 0.0;                          //initial values
    v[0] = 1.0;

    for (int i = 0; i < N-1; i++)
    {
        x[i+1] = x[i] + v[i] * dt;
        v[i+1] = v[i] -  x[i] * dt;      //f=ma=-kx, k/m=1
        cout<<i*dt<<"\t"<<x[i]<<"\t"<<v[i]<<endl;
    }
    return 0;
}
```

Code example

- 1.3.Intro.cpp
- g++ 1.3.Intro.cpp
- ./a.out
- gnuplot
- plot "data.dat" using 1:2, "data.dat" using 1:3



Simulation results: a simple harmonic oscillation

Homework

- An oscillation with resistance
- $F = - k*x - b*v$